<ins>Amendments to the Specification</ins>:

Please amend the Cross-Reference to Related Applications section beginning at page 1, line 4 of the specification as follows:

This application is related to the following applications filed on even date herewith:

1. U.S. Patent Application Serial ~~No. __________, filed~~ <ins>No. 09/629,696, filed</ins> in the names of Yee S. Ng et al, and entitled EDGE ENHANCEMENT OF GRAY LEVEL ~~IMAGE~~ <ins>IMAGES</ins>.

2. U.S. Patent Application Serial ~~No. __________, filed~~ <ins>No. 09/629,994, filed</ins> in the names of Yee S. Ng et al,. and entitled EDGE ENHANCEMENT PROCESSOR AND METHOD WITH ADJUSTABLE STRENGTH OF GRAY LEVEL OUTPUT.

3. U.S. Patent Application Serial ~~No. __________, filed~~ <ins>No. 09/628,397, filed</ins> in the names of Yee S. Ng et al, and entitled EDGE ENHANCEMENT PROCESSOR AND METHOD WITH ADJUSTABLE THRESHOLD SETTING.

4. U.S. Patent Application Serial ~~No. __________, filed~~ <ins>No. 09/630,435, filed</ins> in the names of Hwai-Tzuu Tai et al, and entitled IMAGE RECORDING APPARATUS AND METHOD PROVIDING PERSONALIZED COLOR ENHANCEMENT.

Please amend the paragraph beginning at page 14, line 5 of the original specification as follows. This listing shall replace all prior listings:

A robust implementation of this processing is indicated by the flowchart of FIG. 9 where ~~the pixel having coordinates (x,y)~~ <ins>each pixel 950 in a page 960 (beginning at location x-0, y=0 910)</ins> is mapped <ins>line-by-line 920, 930</ins> to a certain location (I,J) <ins>940</ins> in a brick plane which location is then provided as one input to a halftone screen lookup table <ins>970</ins> that also has input to it the gray value g(x,y) of the pixel. The lookup table <ins>970</ins> stores rendered pixel values for halftone rendering of the image pixel g(x,y). In this example there are 241 x 255 rendering values in the LUT (brick width times number of brick planes). Further reduction of the size of the table can be made by recognizing that gray value 0 and 255 have I and J values that are irrelevant since in this example each pixel having a gray

value of 0 and 255 is rendered at that respective value. In the flowchart of FIG. 9 the pixel image coordinate value x,y is input to a calculator that takes the value of the x-coordinate and adds that to a value of the y coordinate which has been first divided by the brick height and then multiplied by a brick offset value. This sum is then divided by the brick width wherein only the remainder is retained as the brick coordinate value for I. For example, where X=178,Y=1,Bh=1,Bs=177,and Bw=241, the calculation is made of adding 178 + (1/1)177=355, which is then divided by the brick width of 241 to yield a remainder I=114. The J coordinate value is determined by taking the y coordinate value in the image plane and dividing it by the brick height and retaining the remainder as the value for J. In this example for this screen the value of J is always zero, however, as noted above, some screens may have a brick height of two or more and so the J coordinate in the brick plane becomes essential to determine. Implementation of the brick coordinate calculator may be by software as processed by a computer or by a chip that is designed to perform this calculation. The calculation may be expressed by the formula:

$$I = (X +(Y/Bh) * Bs) \%Bw$$

wherein "%" indicates that a division operation is made wherein a remainder is determined. As noted above, Bh in certain situations is equal to one so the equation simplifies in such situation to:

$$I = (X + Y * Bs) \%Bw$$

Please amend the paragraph beginning at page 17, line 25 of the original specification as follows. This listing shall replace all prior listings:

In order to generate rendered screen values for a tile the various tile parameters <u>242</u> such as screen angle, lines per inch, number of gray levels per pixel are considered. In addition the nature of the dot driver <u>241</u> and dot type growth <u>243</u> pattern are also considered. An example of a dot driver is illustrated in Fig. 24 for a 16x16 dot size driver having a circular or spiral type of growth pattern wherein dots in a cell tend to grow from the center outwardly. Other types of dot drivers may be used and suited to other shapes of growth patterns such as growth along a line, or an ellipse. These factors may be input to a dot membership function generator <u>244</u>, which considers cells within a tile and the contribution of neighborhood pixels within that tile. A screen profile builder <u>245</u>

may then be used to determine the total gray level in the tile by summing of the exposure values at pixel locations that are not yet quantized. A screen profile quantizer 246 then quantizes the individual pixel rendered values so that these values can be expressed in a form of a whole number, for example, 0-255 in a system having an eight bits per pixel bit depth, and used by the threshold tile LUT 247.